

Guidance Through Surrogate: Toward a Generic Diagnostic Attack

Muzammal Naseer^{id}, Salman Khan^{id}, *Senior Member, IEEE*, Fatih Porikli^{id},
and Fahad Shahbaz Khan

Abstract—Adversarial training (AT) is an effective approach to making deep neural networks robust against adversarial attacks. Recently, different AT defenses are proposed that not only maintain a high clean accuracy but also show significant robustness against popular and well-studied adversarial attacks, such as projected gradient descent (PGD). High adversarial robustness can also arise if an attack fails to find adversarial gradient directions, a phenomenon known as “gradient masking.” In this work, we analyze the effect of label smoothing on AT as one of the potential causes of gradient masking. We then develop a guided mechanism to avoid local minima during attack optimization, leading to a novel attack dubbed *guided projected gradient attack* (G-PGA). Our attack approach is based on a “*match and deceive*” loss that finds optimal adversarial directions through guidance from a surrogate model. Our modified attack does not require random restarts a large number of attack iterations or a search for optimal step size. Furthermore, our proposed G-PGA is generic, thus it can be combined with an ensemble attack strategy as we demonstrate in the case of auto-attack, leading to efficiency and convergence speed improvements. More than an effective attack, G-PGA can be used as a diagnostic tool to reveal elusive robustness due to gradient masking in adversarial defenses.

Index Terms—Adversarial attack, gradient masking, guided optimization, image classification, label smoothing.

I. INTRODUCTION

A DEFENSE can cause gradient masking if it does not allow an adversarial attack to calculate useful gradient directions to deceive a model. Papernot *et al.* [1] found that gradient masking alone is not a robust way to devise a well-rounded defense since adversarial perturbations can be discovered for such models using alternative means, e.g., a smooth version of the same model or from a substitute model. In this manner, the same attack can be used to fool the model by intelligently estimating perturbation directions to which the model remains highly sensitive even after deploying the defense.

We hint toward the presence of gradient masking in the recent state-of-the-art defense mechanisms (feature scattering

(FS) [2], adversarial mixup (AvMixup) [3], and mixup inference [4]). These defenses show excellent robustness in the more challenging white-box setting, where all network parameters, model architecture, and training details are known. Specifically, single-step [5], [6] as well as iterative optimization-based [7]–[10] attacks find it difficult to calculate useful gradient directions in order to launch a successful attack. We note that these defenses use *label smoothing* [11], [12] as a regularization measure to create smooth loss surfaces. While learning smooth loss surfaces is a preferable property of robust models [13], the existence of large contagious regions of adversarial examples [5], [14] means the smoothness is only achieved in the small neighborhood of the training manifold. We demonstrate that label smoothing can, thus, cause gradient masking, thereby leading to inflated estimates of model robustness.

In this work, by studying the masking behavior of label smoothing, we characterize a new category of gradient masking which may not be intentionally caused by shattered, randomized, or vanishing gradients. Instead, we suggest the specific “*loss surface*” induced by a loss function and/or the training algorithm can also lead to gradient masking. This means a white-box attack would not be much successful even with all the knowledge about the training process and network parameters. We found that since such masking is quite subtle, it may not be caught by the diagnostic behaviors identified by Athalye *et al.* [15]. Subsequently, we propose a new attack approach (G-PGA) based on the adversarial directions from a surrogate model that acts as a useful test for existing defenses. Our approach utilizes a novel contrastive “*match and deceive*” loss to find harmful directions using guidance from the teacher to deceive the source model (Fig. 1).

Contributions: The main contribution of this work is to study label-smoothing as a case study to highlight the masking behavior of recent popular defenses. We complement our findings with a strong attack method that generalizes across various defense mechanisms to unveil the gradient masking effect. The salient aspects of our approach are as follows.

- 1) *Relation of Label Smoothing With Gradient Masking:* We analyze the effect of label smoothing on adversarial training (AT). To this end, we develop an AT algorithm, called mask-AT, that combines random initialization and label smoothing with a single-step adversarial attack to achieve masking without losing clean accuracy. Our mask-AT highlights label smoothing as the common root cause of elusive robustness of the recent defenses [2], [3].

Manuscript received 30 June 2021; revised 12 February 2022 and 7 May 2022; accepted 19 June 2022. (Corresponding author: Muzammal Naseer.)

Muzammal Naseer and Salman Khan are with the Computer Vision Department, Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates, and also with the College of Engineering and Computer Science, The Australian National University, Canberra, ACT 2601, Australia (e-mail: muz.pak@gmail.com).

Fatih Porikli is with Qualcomm, San Diego, CA 92121 USA.

Fahad Shahbaz Khan is with the Computer Vision Department, Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates, and the Computer Vision Laboratory (CVL), Linköping University, 581 83 Linköping, Sweden.

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TNNLS.2022.3186278>.

Digital Object Identifier 10.1109/TNNLS.2022.3186278

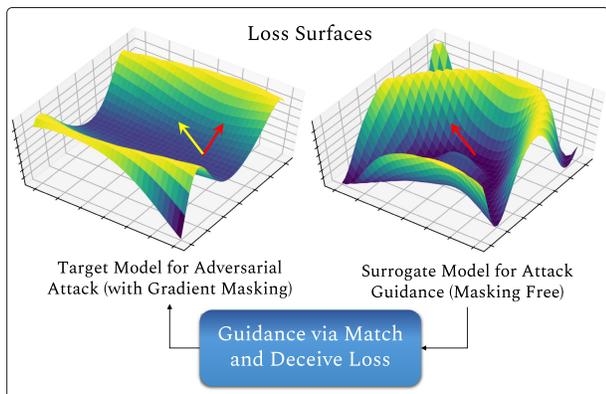


Fig. 1. We analyze the effect of gradient masking on popular defenses and propose a guidance-based attack algorithm, called G-PGA. Our approach seeks guidance from a masking-free surrogate model to find useful attack directions, which, in turn, can be used to diagnose gradient masking.

- 2) *Guiding Mechanism*: We develop a guiding mechanism that allows finding useful gradient directions during attack optimization. The idea is to exploit gradient information from a *masking-free* surrogate model to distill and strengthen the adversarial noise. This way attack optimization avoids being stuck to local minima by observing the correct gradient direction. The masking-free surrogate provides healthy gradient directions [15], e.g., naturally or adversarially trained models (Madry [7] or trades [13]).
- 3) *Match and Deceive Loss*: Our guided attack called G-PGA is based on a novel “*match and deceive*” loss function. The purpose of this loss is to exploit surrogate information based on the principles of rescaling and redirection, achieved by normalized cross-entropy (CE) and contrastive directional objectives, respectively (Section IV).

II. RELATED WORK

A. Adversarial Training

Robust training [7], [13], [16]–[19] constitutes finding adversarial examples by maximizing the model loss and then updating model parameters to correctly classify them. Many robust training methods are proposed, including Madry *et al.* [7] that solved the maximization step with an iterative and computationally expensive attack. Recently, [20] reduced the training cost significantly with a single-step attack [5] combined with better initialization and early termination. Zhang *et al.* [13] proposed to control the tradeoff between clean and adversarial accuracy. Adversarial robustness was further improved in [21] by focusing on the misclassified examples during training. Carmon *et al.* [22] proposed to train on extra unlabeled data that lead to higher robustness with a lower drop in clean accuracy. Zhang and Wang [2] find adversaries by maximizing optimal transport distance and achieving high robustness when combined with label smoothing. Similarly, [3] further enhances the adversarial robustness with minimal loss in clean accuracy. They interpolate between clean and adversarial samples as well as labels with different smoothing factors. Another line of defense uses input processing [23]–[25] to mitigate the adversarial effect.

For example, [4] exploits mixup [26]-based processing along with AT to achieve higher robustness.

B. Gradient Masking

Previous work from Athalye *et al.* [15] noted three causes for obfuscated gradients, namely, shattered gradients, stochastic gradients, and vanishing/exploding gradients. For the first problem, they propose Backward Pass Differentiable Approximation (BPDA), Expectation Over Transformation (EOT) attack for the stochastic/randomized defense and reparametrization and stable optimization for the third category. Furthermore, they reported a number of tests to characterize gradient masking, such as one-step attacks must perform inferior to iterative attacks and black-box attacks should perform lower than white-box attacks. However, the identified behaviors do not form a complete set of possible indicators for gradient masking. In this article, we study state-of-the-art defense approaches [2], [3] and note that although they generally pass the tests prescribed by [15], they are still potentially suffering from gradient masking. This is demonstrated by our extensive experiments that show a significant drop in their performance under our attack.

C. Adversarial Attacks

A number of adversarial attacks [27]–[36] have been proposed to evaluate the robustness of deep neural networks. Among them, projected gradient descent (PGD) [7] and Carlini and Wagner (CW) [30] attacks are computationally feasible and the most popular ones. However, these attacks fail to faithfully estimate the robustness of recent defenses [2], [3] even when they are adaptive to the defense [4]. This has led to the development of new attacks, where [37] introduced a sampling strategy to enhance the performance of PGD and CW, while [38] proposed an auto-attack based on parameter-free objectives along with evaluating the model on an ensemble of attack strategies. Auto-attack depends on a large number of queries, and extra information, such as identifying, which input samples are misclassified by a model and then adapting to the new attack settings for those input samples. In real-world settings, this information may not be available to the attacker.

D. Our Differences

Our proposed guiding mechanism is generic in its nature. When used as a stand-alone attack, it performs equivalent to state-of-the-art auto-attack [38] while being significantly less computationally expensive. It does not require a large number of queries or random restarts. It also does not depend upon extra information, such as misclassification indication for a given sample. Our proposed method complements the current attacks. When our guided mechanism is combined with existing approaches [38]–[40], it allows faster convergence and enhances the performance of an attack, as shown in combination with attack strategies proposed by auto-attack in Section V.

III. GRADIENT MASKING DURING ATTACK OPTIMIZATION: CASE STUDY

It is well-known that gradient masking [15], [41] can cause optimization difficulties during adversary generation,

resulting in inflated robustness. Pang *et al.* [42] show that only moderate use of label smoothing (6) helps in boosting adversarial robustness but excessive label smoothing can decrease the robustness. We note that recent state-of-the-art AT methods [2], [3] use higher label smoothing during training. Therefore, an important question is: *when does label smoothing causes gradient masking?* Here, we analyze the effect of label smoothing on adversarial robustness as a case study, highlighting the need to diagnose such behaviors [43]–[45].

Proposition 1: Consider a model trained using a regular CE loss (ℓ_{ce}) with label smoothing outputs logits $\mathbf{a} \in \mathbb{R}^N$. Then, the gradients used to update the model are relatively weaker, i.e., $\partial \ell_{ce}(\mathbf{a}, \hat{\mathbf{y}})/\partial a_i < \partial \ell_{ce}(\mathbf{a}, \mathbf{y})/\partial a_i$ for $\delta \in (0, 1]$ where $\mathbf{y} \in \mathbb{R}^N$ denote the one-hot encoded labels and $\hat{\mathbf{y}}$ denotes its smoothed version. As a result, smooth loss surfaces are learned close to the training data manifold, thereby suppressing gradients used to craft adversaries in local neighborhoods. This phenomenon causes gradient masking and leads to inflated robustness of the learned model in small neighborhood of the training data [1].

Sketch Proof: For a model being trained with CE loss ℓ_{ce} and one-hot encoded ground-truth labels \mathbf{y} , the gradients are given by

$$\frac{\partial \ell_{ce}(\mathbf{a}, \mathbf{y})}{\partial a_i} = \sigma(a_i) - y_i \quad (1)$$

where

$$\sigma(a_i) = \frac{\exp(a_i)}{\sum_j \exp(a_j)}. \quad (2)$$

In comparison, the gradients for the same model trained with smoothed labels $\hat{\mathbf{y}}$ are given by

$$\frac{\partial \ell_{ce}(\mathbf{a}, \hat{\mathbf{y}})}{\partial a_i} = \sigma(a_i) - \hat{y}_i \quad (3)$$

where

$$\hat{y}_i = \begin{cases} (1 - \delta), & \text{if } y_i = 1 \\ \frac{\delta}{N - 1}, & \text{if } y_i = 0 \end{cases} \quad (4)$$

where N is the total number of classes. The above-mentioned expression shows if $\delta \in (0, 1]$, then whether $y_i = 0$ or $y_i = 1$, the difference with predicted probability score for a class will always be less than the case when nonsmooth labels are used. Hence

$$\frac{\partial \ell_{ce}(\mathbf{a}, \mathbf{y})}{\partial a_i} > \frac{\partial \ell_{ce}(\mathbf{a}, \hat{\mathbf{y}})}{\partial a_i}. \quad (5)$$

Label smoothing leads to smooth loss surfaces during model training. However, the existence of large contagious regions of adversarial pockets in the data manifold [5], [14] means that such smoothness is only achieved close to the training data manifold. Thus, the gradient directions computed from the same model even for the case of white-box adversaries and do not disclose directions to which the model still remains susceptible.

A. Case Study

We design an AT algorithm to showcase how an attack fails in the presence of gradient masking. This masking is

Algorithm 1 Mask-AT

- 1: A batch of benign samples $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$, a model f parameterized by θ , perturbation budget ϵ , scaling parameter η , label smoothing factor δ and CE loss ℓ_{ce} .
 - 2: **for** $i = 1$ to n **do**
 - 3: $\tilde{\mathbf{x}}_i = \mathbf{x}_i + \boldsymbol{\mu}(\eta \cdot \epsilon)$: $\boldsymbol{\mu} \sim \text{Uniform}(-1, 1)$ ▷ Take a random step
 - 4: $\tilde{\mathbf{x}}_i = \text{clip}(\text{FGSM}(\tilde{\mathbf{x}}_i), \mathbf{x}_i - \epsilon, \mathbf{x}_i + \epsilon)$ ▷ Generate adversary
 - 5: **end for**
 - 6: $\theta = \theta - \nabla \ell_{ce}(f(\tilde{\mathbf{x}}), \hat{\mathbf{y}}(\delta))$ ▷ Update the model parameters
-

introduced by label smoothing and we call the resulting AT algorithm as mask-AT. Mask-AT is based on a single-step adversarial attack known as Fast Gradient Sign Method (FGSM) [5] combined with larger random initialization and label smoothing. Our AT (Algorithm 1) takes a large random step in the input space by adding a uniform noise to a given sample and then adversarial examples are computed by taking a single-adversarial step using FGSM. Model parameters are updated by minimizing the empirical loss (CE) with smooth labels on these adversarial examples. We observe that such simple AT shows better robustness against well-studied iterative attacks, such as PGD, while maintaining a high clean accuracy (Fig. 2). This behavior resonates well with the recent state-of-the-art defenses, including FS [2] and AvMixup training [3]. Thus, our experiment sheds a light on how gradient masking introduced by label smoothing can play a significant part in achieving high adversarial robustness.

B. Mask-AT Training

To study the effect of label smoothing, we train a ResNet18 on CIFAR10 using Mask-AT (Algorithm 1). Models are trained using SGD optimizer for 200 epochs with batch-size 60. Pixel values are scaled to $[-1, +1]$. Learning rate is set to 0.1 and decreased by a factor of 10 at epochs 60 and 90. Perturbation budget, ϵ , is set to $8/255$ during training. Label smoothing [2] is performed as

$$\hat{\mathbf{y}} = \left(1 - \delta - \frac{\delta}{N - 1}\right) * \mathbf{y} + \frac{\delta}{N - 1} \quad (6)$$

where δ is label smoothing factor, N represents number of classes, and $\mathbf{y} \in \mathbb{R}^N$ is the one-hot encoded label. Applying (6) to \mathbf{y} reduces the confidence of true class to $1 - \delta$.

C. Mask-AT Evaluation

We evaluate ResNet18's robustness against PGD attack [7] with 20 iterations and step of $(2/255)$.

D. Analysis and Observations

We dissect each component of Mask-AT (Algorithm 1) in order to better understand the role of label smoothing, random initialization, and attack iterations. Results presented in Fig. 2 can be analyzed as follows.

- 1) *Effect of Random Initialization:* We set $\delta = 0$, i.e., no label smoothing is applied. We perturb the input

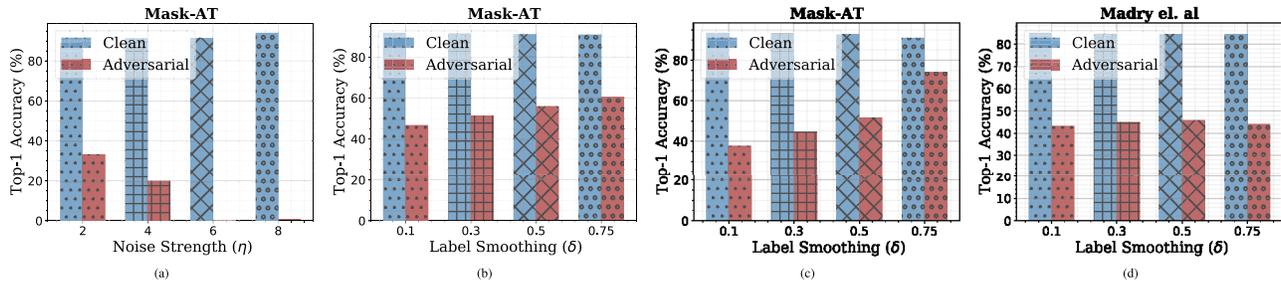


Fig. 2. *Analysis on Gradient Masking*: Clean versus adversarial accuracy (% Top-1) of ResNet18 on CIFAR10 is reported against PGD attack with 20 iterations. PGD successfully highlights true robustness in the absence of gradient masking, e.g., when models are trained on FGSM: (a) or using Madry’s method [7] (d). However, PGD erroneously indicates very high robustness of models trained on FGSM adversaries with label smoothing (b) and (c). Thus, with the help of label smoothing only, adversarial robustness of Mask-AT (Algorithm 1) increases from zero to around 75% while maintaining the clean accuracy [plot (c) $\eta = 6$ and $\delta = 0.75$]. This signifies how the use of a certain component during AT (such as label smoothing in this case) can paralyze an attack and shows fake robustness. A solution to this problem is to guide the attack optimization (Section IV). (a) Effect of η with $\delta = 0$. (b) Effect of δ with $\eta = 4$. (c) Effect of δ with $\eta = 6$. (d) Effect of δ with $\eta = 4$.

sample with uniform noise before running the FGSM attack during training (Algorithm 1). The strength of uniform noise is controlled by η . The higher the η , the larger the random step taken before the attack. We observe in Fig. 2(a) that model clean accuracy increases by increasing η but at the same time, its adversarial robustness decreases. This signifies that training a model with no label smoothing on adversaries computed using FGSM does not make the model robust and it also does not introduce any gradient masking as PGD [7] successfully exposes weak robustness. This behavior, however, changes as we introduce label smoothing during training.

- 2) *Effect of Label Smoothing*: We now fix the value of η in the next experiment, as shown in Fig. 2(b) and (c), and start increasing the label smoothing factor, δ (6). We observe that the higher the value of δ , the higher the robustness of the model against PGD attack. This is an abnormal behavior such that the robustness of ResNet18 trained using Algorithm 1 at $\eta = 6$ goes from 0% to 75% just by increasing label smoothing, δ .
- 3) *Effect of Attack Iterations*: Finally, we study if such gradient masking effect continues with iterative attack training as well. We fix the value of η and train the model by running ten attack iterations, rather than on FGSM. This is equivalent to Madry *et al.* [7]. We observe that the masking effect caused by the combination of label smoothing and random noise reduces significantly. This indicates that such masking phenomena prevail in models trained with label smoothing and single-step attacks, such as [2] and [3].

E. Outcome and Motivation for Guided Optimization

Our experimental analysis [Fig. 2(a)–(d)] shows interesting insights about gradient masking. We observe that the same attack (PGD) works perfectly well when there is no gradient masking effect. For example, PGD works as expected when models are trained using FGSM with no label smoothing or Madry’s method (with or without label smoothing) but it fails on models trained using FGSM combined with large random initialization and label smoothing. Such uncertainty in attack optimization leads to elusive robustness [2]–[4] and there is

a need for the adversarial attack that performs consistently. We provide a complimentary approach to the previous attack solutions [15], [38] by proposing to guide optimization using gradient direction from a surrogate model thus avoiding local minima due to gradient masking.

IV. GUIDED PROJECTED GRADIENT ATTACK

In this section, we develop a guiding mechanism based on a new “*match and deceive*” loss that can quickly expose gradient masking with a small number of attack iterations. This leads us to a novel attack named *guided projected gradient attack* (G-PGA) attack. (Algorithm 2).

A. Need for External Guidance

A *white-box adversary* is created using the full knowledge of model architecture, pretrained weight parameters, training loss along with any randomness used to perturb the input samples during training. Consider a model f parameterized by θ is adversarially trained using a given training mechanism, train_f (e.g., [2], [3]). If train_f leads to gradient masking (as shown in Section III) then white-box adversaries will be less effective against the trained model.

We observe that the guidance from the loss function can play a key role in launching a successful attack. For example, the adversary is more effective when computed using CE loss for over-confident models but becomes less effective on models trained using aggressive label smoothing (Section III). Since the guidance available in the white-box settings (with gradient masking) is nonconductive to finding reliable attack directions, the attack needs to look “*elsewhere*” for better guidance. To this end, we propose to introduce a surrogate model in the attack pipeline which is used to find optimal attack directions on the source network.

A naive strategy would be to use a surrogate model, h , parameterized by ϕ trained using a masking-free method (train_h) (e.g., naturally training using CE or AT using [7], [13]) with a similar architecture as model f . Adversaries can be computed against model h and then transferred to the model f . The problem with this approach is that these adversaries contain gradient noise specific to the model parameters ϕ trained using a given approach (train_h) leading to suboptimal results

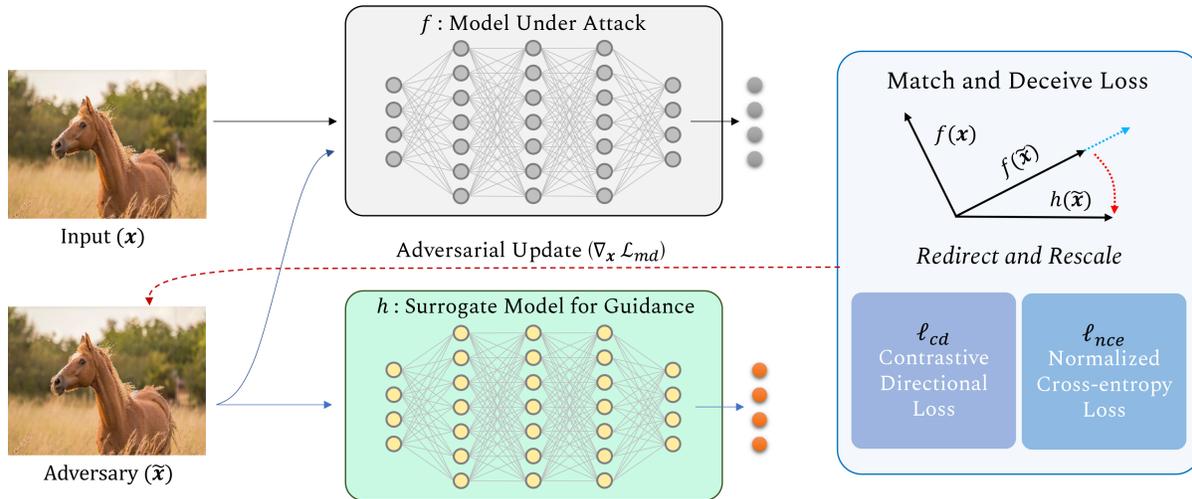


Fig. 3. G-PGA is based on match and deceive loss which uses information provided by the surrogate model to redirect and attentively rescale the logits of the under attack model in order to overcome gradient masking and find optimal adversarial direction.

for the model f (see Section VI for analytical insights). Note that the adversaries created using model, h , are considered *black-box* to the model f , if $\text{train}_f \neq \text{train}_h$ even when f and h share similar architecture.

B. Match and Deceive Loss

Our proposal is to use masking-free information from the surrogate model, h , as a guide to finding an optimal adversarial direction for the model f . The objective is to avoid the white-box setting with potential optimization difficulties and the black-box scenario, as both lead to weaker adversaries. We achieve this guiding mechanism (Fig. 3) as follows.

- 1) *Redirection*: During the attack, redirect the optimizer with a supervisory signal that matches the adversarial directions from the output space of model f , with the surrogate model h . Here, the intuition is that if h is masking-free then the optimizer should move along the adversarial direction defined by h . Redirection is achieved using a contrastive directional loss (7).
- 2) *Rescaling*: Attentively rescale the logit space outputs of the under-attack model using the guidance of logits from the surrogate model. Rescaling allows us to incorporate masking-free information from the surrogate into the CE loss. This is achieved by a normalized CE loss (9).

1) *Contrastive Directional Loss*: Consider a benign input sample \mathbf{x} , an adversarially perturbed sample $\tilde{\mathbf{x}}$, then contrastive directional loss ℓ_{cd} minimizes the similarity between the output vectors $f(\mathbf{x}) \in \mathbb{R}^k$ and $f(\tilde{\mathbf{x}}) \in \mathbb{R}^k$ in the k -dimensional logit-space, a desirable property for an optimal adversary to maximally perturb the input. It simultaneously maximizes the similarity between $f(\tilde{\mathbf{x}})$ and $h(\tilde{\mathbf{x}}) \in \mathbb{R}^k$, which provides better adversarial gradient direction considering h is adversarially trained and masking free. The ℓ_{cd} loss is defined as

$$\ell_{cd} = -\log \frac{\exp(\mathcal{S}(f(\tilde{\mathbf{x}}), h(\tilde{\mathbf{x}})))}{\exp(\mathcal{S}(f(\mathbf{x}), f(\tilde{\mathbf{x}}))) + \exp(\mathcal{S}(f(\tilde{\mathbf{x}}), h(\tilde{\mathbf{x}})))} \quad (7)$$

where $\mathcal{S}(\cdot)$ represents the cosine similarity between two given vectors, i.e., $\mathcal{S}(\mathbf{a}, \mathbf{b}) = ((\mathbf{a}^\top \mathbf{b}) / (\|\mathbf{a}\| \|\mathbf{b}\|))$.

Algorithm 2 Guided Projected Gradient Attack

- 1: A benign sample \mathbf{x} , a classifier f , a surrogate model h , perturbation budget ϵ , number of attack iterations T , step size κ .
- 2: $\mathbf{g}_0 = \mathbf{0}$; $\tilde{\mathbf{x}} = \mathbf{x}$, $t \leftarrow 0$;
- 3: **repeat**
- 4: $t \leftarrow t + 1$;
- 5: Forward pass $\mathbf{x}, \tilde{\mathbf{x}}$ through f and h and compute $f(\mathbf{x})$, $f(\tilde{\mathbf{x}})$ and $h(\tilde{\mathbf{x}})$.
- 6: Compute contrastive directional loss ℓ_{cd} (Eq. 7) using $f(\mathbf{x})$, $f(\tilde{\mathbf{x}})$ and $h(\tilde{\mathbf{x}})$.
- 7: Compute normalized cross-entropy loss ℓ_{nce} (Eq. 9) using $f(\tilde{\mathbf{x}})$ and $h(\tilde{\mathbf{x}})$.
- 8: Compute the match and deceive loss \mathcal{L}_{md} (Eq. 10).
- 9: Backward pass and compute gradients $\mathbf{g}_t = \nabla_{\mathbf{x}} \mathcal{L}_{md}$.
- 10: Use gradients to update perturbation estimate,

$$\tilde{\mathbf{x}}_{t+1} = \tilde{\mathbf{x}}_t + \kappa \cdot \text{sign}(\mathbf{g}_t).$$

- 11: Project the adversary within allowed perturbation budget, ϵ

$$\tilde{\mathbf{x}}_{t+1} = \text{clip}(\tilde{\mathbf{x}}_{t+1}, \mathbf{x} - \epsilon, \mathbf{x} + \epsilon).$$

- 12: **until** $t \leq T$

2) *Normalized Cross-Entropy*: As $f(\tilde{\mathbf{x}})$ and $h(\tilde{\mathbf{x}})$ represent the logit response for the perturbed sample $\tilde{\mathbf{x}}$, then guidance through logit rescaling is defined as

$$\mathbf{I}(f(\tilde{\mathbf{x}}), h(\tilde{\mathbf{x}})) = \frac{f(\tilde{\mathbf{x}}) \circ h(\tilde{\mathbf{x}})}{\|f(\tilde{\mathbf{x}})\|_2} \quad (8)$$

where \circ denotes Hadamard product. Adversarial perturbation can be created by maximizing the following normalized CE loss:

$$\ell_{nce} = -\sum_{j=1}^k y_j \log(\sigma(\mathbf{I}(f(\tilde{\mathbf{x}}), h(\tilde{\mathbf{x}}))_j)) \quad (9)$$

where k represents the number of classes, σ is a softmax function, and $\mathbf{y} \in \mathbb{R}^k$ is the corresponding one-hot encoded

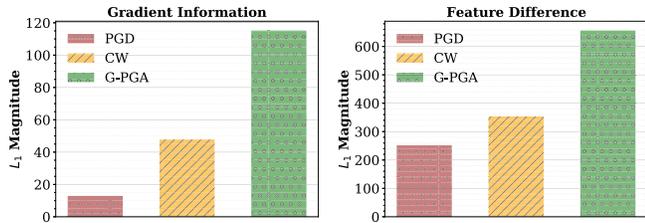


Fig. 4. L_1 magnitude averaged over 10k CIFAR10 test samples (*higher is better*). *Left*: our attack increases the gradient information vital for creating effective adversaries. *Right*: L_1 distance (higher the better) between adversarial and clean features (extracted before the logit layer).

ground-truth vector. The ℓ_{nce} loss promotes both $f(\tilde{\mathbf{x}})$ and $h(\tilde{\mathbf{x}})$ to agree on the misclassification of the perturbed sample with a similar confidence.

The final match and deceive loss is given as follows:

$$\mathcal{L}_{md} = \ell_{nce} - \ell_{cd}. \quad (10)$$

The above loss is maximized to obtain optimal adversaries in our proposed guided attack.

C. Gradient Analysis

For the sake of brevity, we consider $\mathbf{u} = f(\tilde{\mathbf{x}})$, $\mathbf{v} = f(\mathbf{x})$, and $\mathbf{z} = h(\tilde{\mathbf{x}})$. Let us assume their corresponding normalized versions are denoted as $\mathbf{u}' = \mathbf{u}/\|\mathbf{u}\|$, $\mathbf{v}' = \mathbf{v}/\|\mathbf{v}\|$ and $\mathbf{z}' = \mathbf{z}/\|\mathbf{z}\|$. Then, the loss function ℓ_{cd} can be written in terms of dot-product as follows:

$$\ell_{cd} = -\log \frac{\exp(\mathbf{u}' \cdot \mathbf{z}')}{\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}')}. \quad (11)$$

In the following, we compute the gradients of contrastive directional loss function. Since, our goal is to learn the adversarial image $\tilde{\mathbf{x}}$, we are mainly interested in the gradients $(\partial \ell_{cd} / \partial \mathbf{u})$ and $(\partial \ell_{cd} / \partial \mathbf{z})$ involving the base model and surrogate model, respectively. These gradients are given by (proof in the Appendix)

$$\frac{\partial \ell_{cd}}{\partial \mathbf{u}} = -\frac{(\mathbf{z}' - \mathbf{v}') \exp(\mathbf{v}' \cdot \mathbf{u}') (\mathbf{I} - \mathbf{u}' \cdot \mathbf{u}'^T)}{\|\mathbf{u}\| (\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}'))} \quad (12)$$

$$\frac{\partial \ell_{cd}}{\partial \mathbf{z}} = -\frac{\mathbf{u}' \exp(\mathbf{v}' \cdot \mathbf{u}') (\mathbf{I} - \mathbf{z}' \cdot \mathbf{z}'^T)}{\|\mathbf{z}\| (\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}'))}. \quad (13)$$

Fig. 4 shows the empirical analysis for the gradient information provided by the match and deceive loss. We run PGD, CW, and G-PGA (our) attacks for 20 iterations against the mask-AT model and compare the average magnitude of gradients across attacks. We also show the feature distortion caused by the three attacks on CIFAR10. Our results show that the proposed loss can provide stronger gradients with guidance from the surrogate model.

V. EVALUATIONS

A. Experimental Protocols

We evaluate the effect of guided attack optimization on different AT methods, including Madry *et al.* [7], trades [13], FS [2], AvMix [3], and mask-AT (Algorithm 1). Pre-trained model (CIFAR10) for FS is publicly available. Rest of the models are reproduced using open source code

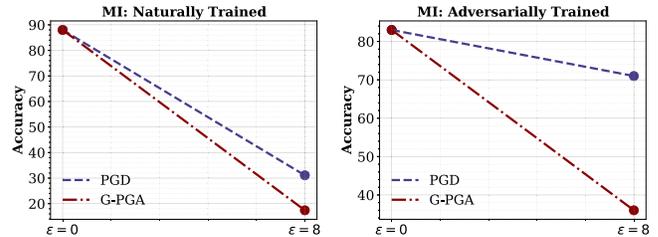


Fig. 5. Perturbations found using our attack break MI [4] without adapting to the defense. Top-1 (%) accuracy is reported on 1k CIFAR10 test samples with WideResNet (*lower is better*). MI is performed with a combination of predicted and other labels as in [48].

bases [2], [3], [13]. We used open source implementation of attacks, including auto-attack [38], FAB [40], and square attack [39]. Training and evaluation are performed on four commonly used datasets: CIFAR10, CIFAR100, Street View House Numbers (SVHN), and ImageNet and we report Top-1 test set accuracy (%). All experiments are conducted using NVIDIA Tesla-V100 with PyTorch library [46]. The adversarial perturbations are l_∞ bounded and clearly mentioned in each experiment. We now evaluate our proposed attack under two settings: 1) *standalone efficiency of G-PGA*; 2) *effect of G-PGA in ensemble of attacks*; and 3) *effect of guidance to noisy gradient estimation*.

B. Standalone Efficiency of G-PGA

We compare our attack strength with baseline methods, including PGD [7] and CW. CW attack is based on a margin loss [30] used by FS [2] and AvMix [3] in PGD attack in a restricted perturbation setting. Step size is set to $2/255$ for all the attacks. All experiments in Table I are conducted on WideResNet. Following insights emerge from our experiments: 1) Our attack performs significantly better than PGD and CW while having 50% fewer iterations. It decreases the robustness of FS [2], AvMix [3] and mask-AT significantly ($<2\%$) as the number of classes increases, e.g., on the CIFAR100 dataset (Table I) and 2) Madry *et al.* [7] and trades [13] are least affected by the masking effect of label smoothing but our attack remains stronger against these defenses.

Breaking Mixup-Inference (MI): Mixup [26] is another way to smooth model output predictions. Pang *et al.* [4] proposed to break the locality of adversarial examples by exploiting the global linear behavior of the model after Mixup training (natural or adversarial). They propose to mix an adversarial sample before inference to reduce attack strength. However, our proposed perturbations (Algorithm 2) significantly reduce the effect of such dynamic inference which reflects our attack's strength (see Fig. 5). It is important to note that the attack is not adapted specifically to MI defense *i.e.* we are able to break the model without any knowledge about the MI defense.

C. Effect of G-PGA in Ensemble of Attacks

Croce and Hein [38] deploy multiple attacks, including modified versions of PGD [7], FAB [40], and query-based square [39] attack. Our proposed attack is computationally less expensive, e.g., 85% less costly to run on CIFAR100 in comparison with the cheaper version of auto-attack [38] (see

TABLE I

EVALUATION (% TOP-1 ACCURACY) OF AT MECHANISMS (*Lower Is Better*). OUR PROPOSED ATTACK (G-PGA, ALGORITHM 2) EFFICIENTLY EXPOSES ANY ELUSIVE ROBUSTNESS WITHIN A FEW ATTACK ITERATIONS

CIFAR10. Perturbation budget is 8/255 in ℓ_∞ norm.											
Defense	δ	Clean	PGD			CW			Proposed Attack		
			10	20	100	10	20	100	10	20	50
Madry <i>et al.</i> [7] (ICLR' 18)	0.5	86.2	49.3	47.5	46.8	47.8	46.4	45.8	46.6	45.2	44.7
Trades [13] (ICML' 19)	0.5	86.0	54.3	53.4	53.1	52.6	52.1	52.0	53.4	52.5	51.9
FS [2] (NeuRIPS' 19)	0.5	90.0	70.9	70.5	68.6	62.6	62.4	60.6	47.0	43.5	40.9
AvMix [3] (CVPR' 20)	0.5-0.7	93.2	73.9	72.1	70.4	65.4	62.5	59.7	57.5	51.5	46.4
Mask-AT (Ours)	0.75	93.4	75.1	74.5	72.9	71.4	68.4	64.6	41.2	34.7	29.7
CIFAR100. Perturbation budget is 8/255 in ℓ_∞ norm.											
Madry <i>et al.</i> [7] (ICLR' 18)	0.5	60.2	26.6	25.9	25.6	24.1	23.7	23.6	25.4	24.9	23.4
Trades [13] (ICML' 19)	0.5	59.1	28.7	28.4	28.3	26.1	25.8	25.7	27.4	26.1	25.4
FS [2] (NeuRIPS' 19)	0.5	74.7	43.0	42.9	42.0	26.5	25.8	24.6	2.2	1.5	1.1
AvMix [3] (CVPR' 20)	0.5-0.7	74.5	45.4	44.8	43.2	31.2	29.7	27.4	4.4	3.0	1.9
Mask-AT (Ours)	0.75	74.2	42.2	41.2	39.5	28.6	27.1	25.2	2.9	1.9	1.6
SVHN. Perturbation budget is 8/255 in ℓ_∞ norm.											
Madry <i>et al.</i> [7] (ICLR' 18)	0.5	96.1	52.1	51.5	51.3	49.2	48.6	47.0	49.0	47.4	46.3
Trades [13] (ICML' 19)	0.5	95.0	62.0	61.8	61.5	58.9	56.7	56.1	58.2	57.3	55.9
FS [2] (NeuRIPS' 19)	0.5	96.0	60.6	46.1	25.7	53.4	37.7	19.3	48.4	34.5	19.0
AvMix [3] (CVPR' 20)	0.5-0.7	96.7	77.2	66.0	40.5	73.2	61.5	37.3	64.5	53.7	36.0
Mask-AT (Ours)	0.75	96.5	74.2	66.1	52.6	70.3	60.4	46.6	54.1	44.6	34.4

TABLE II

OUR PROPOSED ATTACK, G-PGA, SIGNIFICANTLY ENHANCES ATTACK SUCCESS RATE ALONG WITH CONVERGENCE WHEN USED IN COMBINATION OF ATTACKS PROPOSED BY [38]. THESE RESULTS COMPLIMENT GUIDED OPTIMIZATION PROVIDED BY G-PGA. "Q" REPRESENTS NUMBER OF QUERIES SENT TO THE MODEL TO ESTIMATE GRADIENTS. TOP-1 (%) ACCURACY IS REPORTED (*Lower Is Better*)

Defense	Effect of our G-PGA on Auto-Attack								
	CE	CE+Ours	DLR	DLR+Ours	Croce <i>et al.</i> [40]	[40]+Ours	Andriushchenko <i>et al.</i> [39]	[39]+Ours	
	Cost \rightarrow	100	20	100	20	100	50	5k(Q)	10+1k(Q)
FS (CIFAR10)	64.2	42.7	48.9	41.7	40.84	38.9		59.12	41.2
FS (CIFAR100)	44.8	1.01	3.0	0.71	8.34	0.74		24.7	1.0
FS (SVHN)	24.5	12.3	20.2	8.4	15.8	7.9		65.7	43.5

Table IV) but it performs on par to [38] without the need for large number of iterations, random restarts or thousands of queries. However, the guiding mechanism we presented is generic in its nature. Therefore, it can be used in combination with other attacks as proposed by [38]. Hence, we study the effect of G-PGA in combination to each of the four untargeted attacks proposed by [38], including PGD variants based on CE and DLR losses [38], FAB [40], and query-based square attack [39]. We observe in Table II that our guided mechanism enhances the efficiency of each component of auto-attack while decreasing their computational cost. To highlight an example, when combined with G-PGA, square attack decreases the accuracy of the model trained using [2] on CIFAR100 from 24% to 1% within only 1k queries.

D. Effect of G-PGA Against No Label Smoothing

Here, we evaluate different defenses that are not dependent on label smoothing during training. These defense methods, such as Madry *et al.* [7] and trades [13], do not suffer from masking so the performance simple PGD [7], auto-attack [38] and G-PGA is equivalent. Results are presented in Table V. We note simple attacks, such as PGD [7], are effective against defenses that do not suffer from the masking effect, however, our guided attack can expose true robustness even when the model suffers from the gradient masking effect, thus leading

to more reliable robust evaluation. This further highlights that when large label smoothing ($\delta = 0.5$) is used with iterative AT [7], [13], it does not introduce gradient masking but can reduce the adversarial robustness of a model. As an example, adversarial robustness of trades [13] reduces from 52.8 at $\delta = 0.0$ (Table V) to 51.9 at $\delta = 0.5$ (Table I).

We further validate our approach on the large-scale ImageNet [49] dataset. Our approach consistently produces favorable results while being computationally efficient (Tables III and IV). We evaluated publicly available adversarially trained models [47] on a subset of ImageNet (5k samples) against different attacks. The surrogate model used in our G-PGA attack is simply a naturally trained ResNet50 model which is also publicly available [46]. These results show the generalizability of our method across datasets and different surrogate models as well.

E. Guidance to Noisy Gradient Estimation

One strong feature of our proposed guidance is the masking free attack optimization. We empirically validate this hypothesis by observing if our method allows finding useful adversarial directions for query-based adversarial attacks when the gradient estimation becomes unreliable from the original model (Fig. 6). In this case, the attacker has access to only the model's output and needs to estimate the gradients.

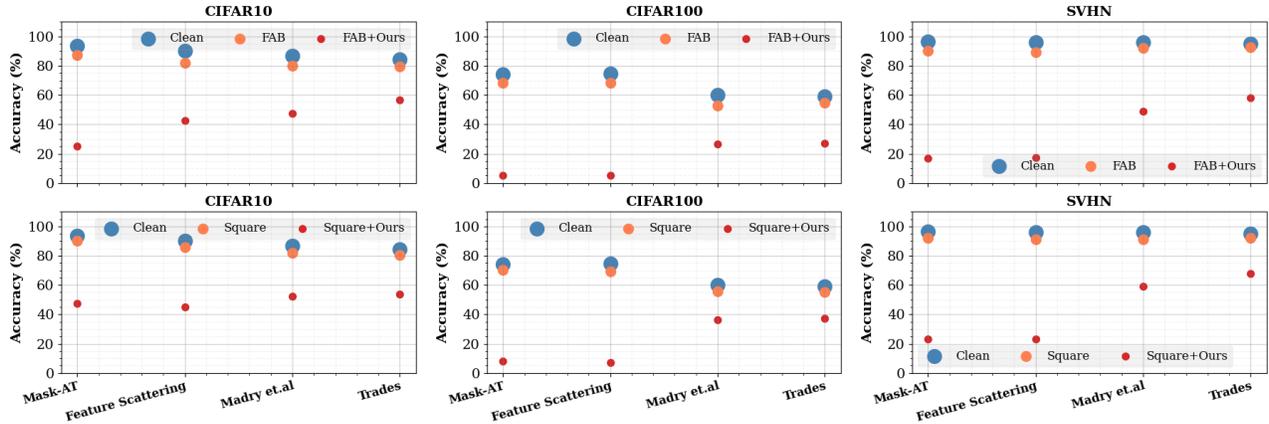


Fig. 6. *Guidance to Noisy Gradient Estimation*: Boundary-based attack FAB [40] (top row) and query-based square attack [39] (bottom row) failed to estimate adversarial direction when adversarially trained models are protected by a dynamic defense. The original (under attacked) model perturbs the incoming input sample \mathbf{x} (clean or adversarial) with uniform noise $(\mathbf{x} + \mu(\eta \cdot \epsilon))$, where $\eta = 1$, $\epsilon = 8$ during inference. We use 5000 queries for the square attack and the FAB attack ran for 100 iterations. Note that under-attacked defenses are trained on adversarial examples with the same perturbation budget of $\epsilon = 8$. Therefore, such dynamic defense causes a minimal drop in clean accuracy, however, these attacks failed mainly because of noisy and suboptimal gradient estimation. Our guiding mechanism allows the same attacks [39], [40] (top and bottom row, respectively) to estimate the optimal adversarial direction with the help of the surrogate model and quickly breaks the original model. Top-1 (%) accuracy is reported on the test sets of each dataset (*lower is better*). Results are the average of five runs.

TABLE III

EFFECTIVENESS OF G-PGA ON ImageNet: WE EVALUATE ADVERSARIALLY TRAINED RESNET50 MODELS FROM [47] AGAINST DIFFERENT ATTACKS INCLUDING G-PGA (OURS). NOTE THAT EVALUATION PERTURBATION BUDGET FOR EACH MODEL IS THE SAME AS ITS TRAINING PERTURBATION BUDGET, E.G., RESNET50 TRAINED ON $\epsilon = 2.0$ IS EVALUATED WITH THE PERTURBATION BUDGET OF $\epsilon = 2.0$. THE SURROGATE MODEL USED IN G-PGA IS A NATURALLY TRAINED RESNET50. G-PGA PERFORMS FAVORABLY IN COMPARISON WITH AUTO-ATTACK WHILE BEING COMPUTATIONALLY EFFICIENT (TABLE IV)

Dataset	Attack	ResNet50		
		$\epsilon = 2.0$	$\epsilon = 4.0$	$\epsilon = 8.0$
ImageNet-5k	PGD	67.6	50.7	32.1
	Auto-Attack	65.6	47.5	30.0
	G-PGA	64.7	47.3	30.0

TABLE IV

COMPUTATIONAL TIME (MINUTES, *Lower Is Better*) IS NOTED ON A TESLA-V100. ATTACKS RAN ON l_∞ ADVERSARIALLY TRAINED MODELS AT $\epsilon = 8$. WIDERESNET IS USED FOR CIFAR10, CIFAR100, AND SVHN DATASETS WHILE RESNET50 IS USED FOR IMAGENET-5K. RESULTS ARE REPORTED USING THE TEST SAMPLES FOR EACH DATASET WITH A BATCH SIZE OF 100

Attack	CIFAR10	CIFAR100	SVHN	ImageNet-5k
Auto-Attack	682.5	2598.2	1774.5	12.8×10^3
Auto-Attack (Cheap)	148.5	581.0	388.7	4252
Proposed Attack	87.0	88.2	226.2	240

Auto-attack [38] also relies on query-based square [39] and boundary-based FAB [40] attacks to estimate the gradients. However, defending against such attacks [39], [40] by injecting noise into the input image or the model's output has been motivated and well-studied in [50] and [51]. These defenses protect the model by corrupting either the input image [50] or the model outputs (logits) [51] with the random noise and thereby corrupting the gradient estimation. Since input and output signal from the model is corrupted at each query during

TABLE V

PERFORMANCES COMPARISON OF DIFFERENT ATTACKS INCLUDING PGD, AA-FULL, AND G-PGA AGAINST DIFFERENT DEFENSES. NO LABEL SMOOTHING IS USED DURING TRAINING OF THESE DEFENSES THAT IS $\delta = 0$. TOP-1 (%) ACCURACY IS REPORTED ON THE TEST SETS OF EACH DATASET (*Lower Is Better*.)

Dataset	Defense	PGD	AA-Full	G-PGA
CIFAR10	Madry <i>et al.</i> [7]	46.3	44.0	44.0
	Trades [13]	56.4	53.1	52.8
CIFAR100	Madry <i>et al.</i> [7]	25.3	24.0	23.0
	Trades [13]	27.9	25.1	25.1
SVHN	Madry <i>et al.</i> [7]	50.7	46.0	46.0
	Trades [13]	60.9	56.8	55.0

attack optimization, therefore such practical attacks [39], [40] struggle to adapt to the deployed defense and cannot find optimal adversarial directions.

As demonstrated in Table II, FAB attack can reduce the robustness of FS [2] to 40.84% but fails when FS is further protected with random noise defenses [50], [51] (Fig. 6). The same observations can be made for the square attack. These findings are consistent across different datasets (CIFAR10, CIFAR100, and SVHN) and training approaches (mask-AT (ours), FS [2], Madry *et al.* [7], and trades [13]) (Fig. 6).

Our proposed guidance compliments these attacks to adapt to such dynamic defenses and fools the under attacked model within a few queries [39] or iterations [40] (Fig. 6). The results of this new experiment shed light on how the masking free guidance through a surrogate model allows an attack to avoid being stuck in a nonoptimal solution even when gradient estimation from the under-attacked model is suboptimal.

VI. ABLATIVE ANALYSIS

A. Attention Visualization of Adversarial Features

Ilyas *et al.* [52] showed that adversarial examples can be explained by the features of the misclassified class labels. We visualize the adversarial features produced with and without our proposed guidance by observing the attention maps

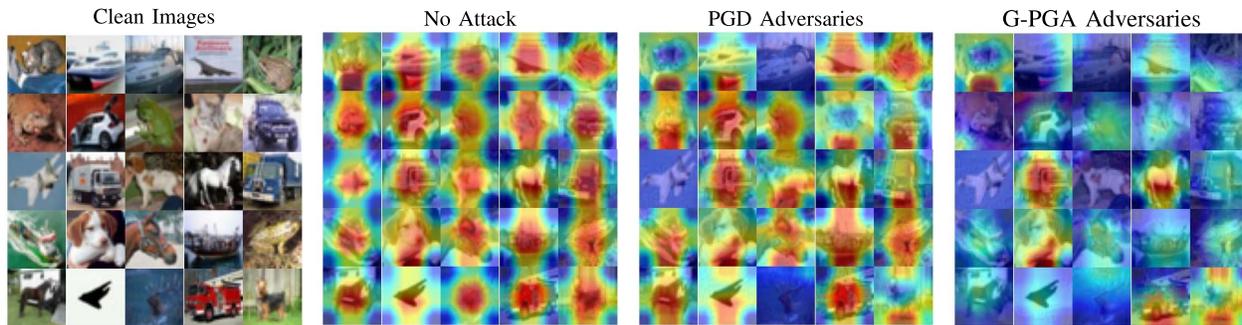


Fig. 7. *Attention Visualization of Adversarial Features*: Adversarial examples can be explained by the L_1 adversarial features imprinted on the original image within a certain perturbation budget [32], [52]. We visualize the presence of the true class features in the input samples using [53]. We observe that adversaries generated by a failed attack (PGD) still contain the features of true class as indicated by the attention maps. G-PGA, on the other hand, successfully maximizes adversarial features and minimizes the presence of true class features as indicated by the dispersed attention. This validates the effectiveness of our approach.

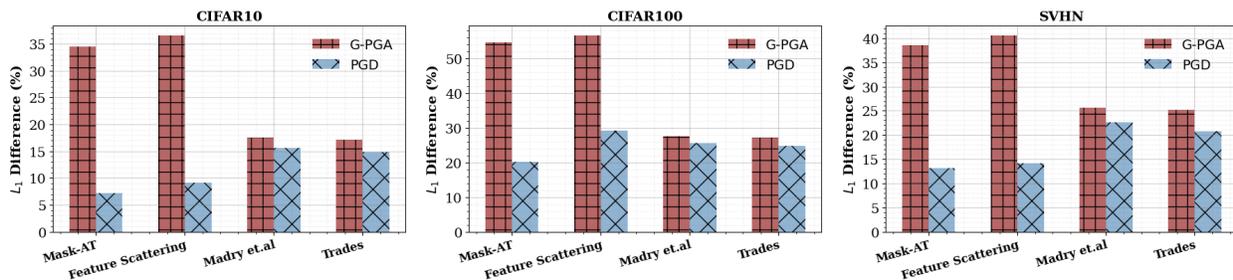


Fig. 8. *Quantify Attention Dispersion*: Our attack approach, G-PGA, minimizes the presence of the features of true class within an allowed perturbation budget (Fig. 7). We quantify this by measuring the L_1 difference (*higher is better*) between attention maps of the true class produced by [53] between adversarial and the clean images. We observe that in comparison with PGD, G-PGA produces a large difference for the models suffering from gradient masking. This highlights that our proposed guidance can successfully reveal gradient masking by minimizing the features of the true class during attack optimization. Results are reported on the test set of each dataset and averaged across the total number of samples.

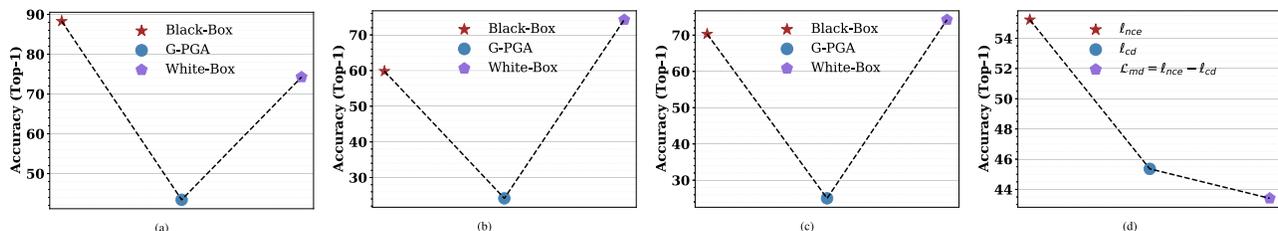


Fig. 9. Accuracy (%) of ResNet18 on CIFAR10 trained using mask-AT (algorithm 1, $\eta = 6$, $\delta = 0.75$) against different attacks. It clearly demonstrates the fooling ability of match and deceives w.r.t. black-box (weak) and white-box (inefficient due to gradient masking) attacks. Results are reported by running PGD attack with 20 iterations with step-size of $(2/255)$ (lower is better). In black-box setting, adversaries are computed on ResNet18 trained using trades [13] with (a) $\beta = 0$, (b) $\beta = 1$, and (c) $\beta = 6$. Finally, plot (d) demonstrates the effectiveness of each component of match and deceive loss (10). (a) Effect of model h ($\beta = 0$). (b) Effect of model h ($\beta = 1$). (c) Effect of model h ($\beta = 6$). (d) Effect of model h ($\beta = 0$).

of the true class using [53]. If an attack is stuck in local minima, then its adversarial features will be weak. Fig. 7 shows attention maps of the adversarial images of a failed attack (PGD) which are closer to the clean images (without any adversarial feature/noise). This means that the attack failed to suppress the features of true class during optimization. On the other hand, attention is dispersed on the adversarial images generated using our guided attack (G-PGA), which indicates that G-PGA successfully maximizes the adversarial features while minimizing the true class features. We quantify this by measuring the L_1 distance between attention maps of the clean and adversarial images generated with and without guidance. Our G-PGA creates more attention dispersion (Fig. 8).

B. Optimal Surrogate

We use the trades [13] framework to find the effective surrogate model. Trades introduces a tradeoff parameter, β .

When $\beta = 0$, trades converge to natural training. When β is increased, the model becomes adversarially stronger at the expense of clean accuracy. We train ResNet18 using [13] at $\beta \in \{0, 1, 6\}$ and use them as surrogate models to observe the surrogate effect on another ResNet18 trained using mask-AT (Algorithm 1 with $\eta = 6$ and $\delta = 0.75$). G-PGA successfully provides the required guidance to overcome gradient masking using surrogate information from naturally ($\beta = 0$) as well as adversarially trained models (Fig. 9). However, for fixed iterations, surrogate information at $\beta = 1$ is the most effective. We further analyze guidance provided by different training methods. Fig. 10 shows that models trained using trades [13] have better black-box transferability and provide faster convergence when used in our attack. However, our approach can successfully exploit surrogate information from naturally trained models as well but requires more attack iterations (Fig. 10). In light of this experiment (Fig. 9), we use the surrogate model trained using [13] at $\beta = 1$. Our analysis

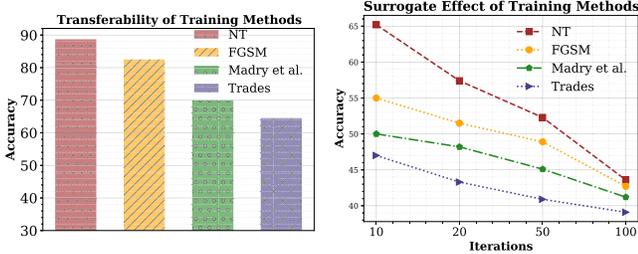


Fig. 10. Robustness of FS [2] in terms of Top-1 accuracy (%) on CIFAR10 dataset. WideResNet is used in the experiment. NT represents natural training. Trades is trained at $\beta = 1$. **Left** plot shows *black-box* robustness of FS against adversarial attack (PGD, ten iterations, step size $2/255$) transferred from models trained using different training methods. **Right** plot shows *white-box* robustness of FS when the different training mechanism used as a surrogate in our proposed attack (Algorithm 2, ten iterations).

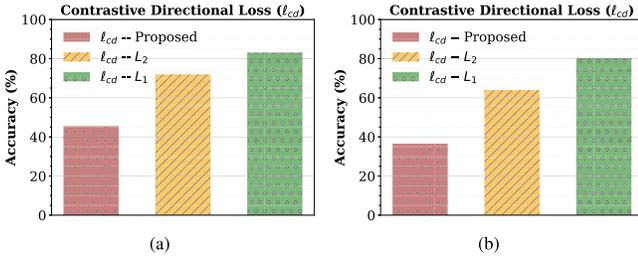


Fig. 11. We evaluate the performance of different metrics within our proposed contrastive loss [l_{cd} , (7)]. Our proposed objective performs significantly better than L_1 or L_2 measures with l_{cd} (*lower is better*). Results are on CIFAR10 test set. (a) Effect of model h ($\beta = 0$). (b) Effect of model h ($\beta = 1$).

TABLE VI

EFFECT OF DIFFERENT SURROGATE MODELS ON G-PGA. LARGE CAPACITY MODELS SUCH WIDERESNET HELPS MORE AS SURROGATE BUT ALSO DIFFICULT TO ATTACK. EVALUATIONS (% TOP-1) ARE PRESENTED AGAINST FS [2] DEFENSE ON CIFAR-10 TEST SET. NiN REPRESENTS "NETWORK IN NETWORKS" MODEL

Original	Surrogate	G-PGA (\downarrow)
WideResNet	ResNet18	36.0
ResNet18	NiN	30.1
ResNet18	WideResNet	28.3

(Fig. 9) shows that the surrogate model trained via trades ($\beta = 1$) is the optimal condition for better guidance but not the necessary condition for an effective guided attack (Table III).

1) *Different Surrogate Models*: Our guided mechanism is not constrained by the same architecture as an original and surrogate model that is G-PGA can achieve guidance from a totally different architecture than the original (under-attacked) model. We present this analysis in Table VI.

2) *Different Contrastive Directional Losses*: When the original model hides gradients or provides noisy gradient estimation then contrastive directional loss pushes the optimizer to move along the adversarial direction defined by the masking free surrogate model. This is demonstrated by our results as well (Tables I and II and Fig. 6). We compare our proposed formulation (7) with L_1 and L_2 losses in contrastive directional loss (Fig. 11). Our proposed objective provides favorable results in contrastive directional loss.

VII. CONCLUSION

Gradient masking is a recurring phenomenon in the evaluation of adversarial robustness. Our work sheds light on the

elusive robustness caused by the label smoothing. We design an AT algorithm that artificially increases model robustness by hiding gradients with the help of label smoothing. We then propose a new attack (G-PGA) based on the concept of guided optimization that exposes gradient masking within a few attack iterations. Our attack approach is based on a novel redirection and rescaling mechanism that uses guidance from a surrogate teacher model on a given target model. G-PGA finds useful adversarial directions that ultimately help to skip local minima during attack optimization. The redirection is achieved with a contrastive directional loss, while rescaling is performed using a normalized CE objective. We hope our findings can act as a guide on the future use of label smoothing in AT and devising diagnostic tools to catch masking.

APPENDIX PROOF OF GRADIENTS

Here, we provide proof for the gradients formulae of l_{cd}

$$\frac{\partial l_{cd}}{\partial \mathbf{u}} = -\frac{(\mathbf{z}' - \mathbf{v}') \exp(\mathbf{v}' \cdot \mathbf{u}') (\mathbf{I} - \mathbf{u}' \cdot \mathbf{u}'^T)}{\|\mathbf{u}'\| (\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}'))} \quad (14)$$

$$\frac{\partial l_{cd}}{\partial \mathbf{z}} = -\frac{\mathbf{u}' \exp(\mathbf{v}' \cdot \mathbf{u}') (\mathbf{I} - \mathbf{z}' \cdot \mathbf{z}'^T)}{\|\mathbf{z}'\| (\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}'))}. \quad (15)$$

Proof:

$$\frac{\partial l_{cd}}{\partial \mathbf{u}} = \frac{\partial \mathbf{u}'}{\partial \mathbf{u}} \cdot \frac{\partial l_{cd}}{\partial \mathbf{u}'} \quad (16)$$

$$\begin{aligned} \frac{\partial l_{cd}}{\partial \mathbf{u}'} &= \frac{\partial}{\partial \mathbf{u}'} \left[-\log \frac{\exp(\mathbf{u}' \cdot \mathbf{z}')}{\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}')} \right] \\ &= -\left(\frac{\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}')}{\exp(\mathbf{u}' \cdot \mathbf{z}')} \right) \\ &\quad \times \frac{\partial}{\partial \mathbf{u}'} \left(\frac{\exp(\mathbf{u}' \cdot \mathbf{z}')}{\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}')} \right) \\ &= -\frac{1}{\exp(\mathbf{u}' \cdot \mathbf{z}') (\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}'))} \\ &\quad \times (\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}')) \frac{\partial}{\partial \mathbf{u}'} \exp(\mathbf{u}' \cdot \mathbf{z}') \\ &\quad - \exp(\mathbf{u}' \cdot \mathbf{z}') \frac{\partial}{\partial \mathbf{u}'} (\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}')) \\ &= -\frac{1}{\exp(\mathbf{u}' \cdot \mathbf{z}') (\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}'))} \\ &\quad \times (\mathbf{z}' \exp(\mathbf{u}' \cdot \mathbf{z}') (\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}')) \\ &\quad - \mathbf{z}' \exp(2 \cdot \mathbf{u}' \cdot \mathbf{z}') - \mathbf{v}' \exp(\mathbf{u}' \cdot \mathbf{z}') \exp(\mathbf{v}' \cdot \mathbf{u}')) \\ &= -\frac{(\mathbf{z}' - \mathbf{v}') \exp(\mathbf{v}' \cdot \mathbf{u}')}{\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}')} \quad (17) \end{aligned}$$

$$\begin{aligned} \frac{\partial \mathbf{u}'}{\partial \mathbf{u}} &= \frac{\partial}{\partial \mathbf{u}} \left(\frac{\mathbf{u}}{\|\mathbf{u}\|} \right) \\ &= \frac{\|\mathbf{u}\| - \mathbf{u} \frac{\partial}{\partial \mathbf{u}} \|\mathbf{u}\|}{\|\mathbf{u}\|^2} = \frac{1}{\|\mathbf{u}\|} - \frac{\mathbf{u} \frac{\partial}{\partial \mathbf{u}} \sqrt{\mathbf{u} \cdot \mathbf{u}^T}}{\|\mathbf{u}\|^2} \\ &= \frac{1}{\|\mathbf{u}\|} (\mathbf{I} - \mathbf{u}' \cdot \mathbf{u}'^T). \quad (18) \end{aligned}$$

Then

$$\frac{\partial l_{cd}}{\partial \mathbf{u}} = -\frac{(\mathbf{z}' - \mathbf{v}') \exp(\mathbf{v}' \cdot \mathbf{u}') (\mathbf{I} - \mathbf{u}' \cdot \mathbf{u}'^T)}{\|\mathbf{u}'\| (\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}'))}. \quad (19)$$

Since, $\mathbf{z} = h(\bar{\mathbf{x}})$ also depends on the adversarial input, we can calculate $(\partial \ell_{cd} / \partial \mathbf{z})$ as follows:

$$\begin{aligned}
 \frac{\partial \ell_{cd}}{\partial \mathbf{z}} &= \frac{\partial \mathbf{z}'}{\partial \mathbf{z}} \cdot \frac{\partial \ell_{cd}}{\partial \mathbf{z}'} & (20) \\
 \frac{\partial \ell_{cd}}{\partial \mathbf{z}'} &= \frac{\partial}{\partial \mathbf{z}'} \left[-\log \frac{\exp(\mathbf{u}' \cdot \mathbf{z}')}{\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}')} \right] \\
 &= - \left(\frac{\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}')}{\exp(\mathbf{u}' \cdot \mathbf{z}')} \right) \frac{\partial}{\partial \mathbf{z}'} \\
 &\quad \times \left(\frac{\exp(\mathbf{u}' \cdot \mathbf{z}')}{\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}')} \right), \\
 &= - \frac{1}{\exp(\mathbf{u}' \cdot \mathbf{z}')(\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}'))} \\
 &\quad \times (\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}')) \frac{\partial}{\partial \mathbf{z}'} \exp(\mathbf{u}' \cdot \mathbf{z}') \\
 &\quad - \exp(\mathbf{u}' \cdot \mathbf{z}') \frac{\partial}{\partial \mathbf{z}'} (\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}')) \\
 &= - \frac{\mathbf{u}' (\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}')) - \mathbf{u}' \exp(\mathbf{u}' \cdot \mathbf{z}')}{\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}')} \\
 &= - \frac{\mathbf{u}' \exp(\mathbf{v}' \cdot \mathbf{u}')}{\exp(\mathbf{v}' \cdot \mathbf{u}') + \exp(\mathbf{u}' \cdot \mathbf{z}')} & (21)
 \end{aligned}$$

which completes the proof.

REFERENCES

- [1] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, Apr. 2017, pp. 506–519.
- [2] H. Zhang and J. Wang, "Defense against adversarial attacks using feature scattering-based adversarial training," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1831–1841.
- [3] S. Lee, H. Lee, and S. Yoon, "Adversarial vertex mixup: Toward better adversarially robust generalization," 2020, *arXiv:2003.02484*.
- [4] T. Pang, K. Xu, and J. Zhu, "Mixup inference: Better exploiting mixup to defend adversarial attacks," in *Proc. Int. Conf. Learn. Represent.*, 2020. [Online]. Available: <https://openreview.net/forum?id=ByxtC2VtPB>
- [5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.
- [6] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," in *Proc. Int. Conf. Learn. Represent. (ICRL)*, 2018.
- [7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Represent.*, 2018. [Online]. Available: <https://openreview.net/forum?id=rJzIBZAb>
- [8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Adversarial examples in the physical world," in *Proc. Int. Conf. Learn. Represent. (ICRL)*, 2017.
- [9] C. Xie *et al.*, "Improving transferability of adversarial examples with input diversity," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2730–2739.
- [10] Y. Dong *et al.*, "Boosting adversarial attacks with momentum," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9135–9193.
- [11] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 4696–4705.
- [12] G. Pereyra, G. Tucker, J. Chorowski, L. Kaiser, and G. Hinton, "Regularizing neural networks by penalizing confident output distributions," 2017, *arXiv:1701.06548*.
- [13] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. El Ghaoui, and M. I. Jordan, "Theoretically principled trade-off between robustness and accuracy," 2019, *arXiv:1901.08573*.
- [14] P. Tabacof and E. Valle, "Exploring the space of adversarial images," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 426–433.
- [15] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," 2018, *arXiv:1802.00420*.
- [16] A. Shafahi *et al.*, "Adversarial training for free!" in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 3358–3369.
- [17] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, and B. Dong, "You only propagate once: Accelerating adversarial training via maximal principle," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 227–238.
- [18] H. Zhang *et al.*, "Towards stable and efficient training of verifiably robust neural networks," 2019, *arXiv:1906.06316*.
- [19] H. Salman *et al.*, "Provably robust deep learning via adversarially trained smoothed classifiers," 2019, *arXiv:1906.04584*.
- [20] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," in *Proc. Int. Conf. Learn. Represent.*, 2020. [Online]. Available: <https://openreview.net/forum?id=BJx040EFvH>
- [21] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu, "Improving adversarial robustness requires revisiting misclassified examples," in *Proc. Int. Conf. Learn. Represent.*, 2020.
- [22] Y. Carmon, A. Raghunathan, L. Schmidt, J. C. Duchi, and P. S. Liang, "Unlabeled data improves adversarial robustness," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 11190–11201.
- [23] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," in *Proc. Int. Conf. Learn. Represent.*, 2018. [Online]. Available: <https://openreview.net/forum?id=SyJ7CIWCb>
- [24] N. Akhtar, J. Liu, and A. Mian, "Defense against universal adversarial perturbations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3389–3398.
- [25] M. Naseer, S. Khan, M. Hayat, F. S. Khan, and F. Porikli, "A self-supervised approach for adversarial robustness," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 262–271.
- [26] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," 2017, *arXiv:1710.09412*.
- [27] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Mar. 2016, pp. 372–387.
- [28] A. Modas, S.-M. Moosavi-Dezfooli, and P. Frossard, "SparseFool: A few pixels make a big difference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9087–9096.
- [29] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Jan. 2019.
- [30] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.
- [31] M. Naseer, S. H. Khan, H. Khan, F. S. Khan, and F. Porikli, "Cross-domain transferability of adversarial perturbations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 12905–12915.
- [32] M. Naseer, S. Khan, M. Hayat, F. S. Khan, and F. Porikli, "On generating transferable targeted perturbations," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 7708–7717.
- [33] M. Naseer, S. Khan, and F. Porikli, "Local gradients smoothing: Defense against localized adversarial attacks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1300–1307.
- [34] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh, "EAD: Elastic-Net attacks to deep neural networks via adversarial examples," in *Proc. AAAI*, 2018, pp. 10–17.
- [35] D. Wu, Y. Wang, S.-T. Xia, J. Bailey, and X. Ma, "Skip connections matter: On the transferability of adversarial examples generated with ResNets," in *Proc. ICLR*, 2020.
- [36] M. Naseer, K. Ranasinghe, S. Khan, F. Khan, and F. Porikli, "On improving adversarial transferability of vision transformers," in *Proc. Int. Conf. Learn. Represent.*, 2022. [Online]. Available: <https://openreview.net/forum?id=D6nH3719vZy>
- [37] Y. Tashiro, Y. Song, and S. Ermon, "Diversity can be transferred: Output diversification for White- and black-box attacks," 2020, *arXiv:2003.06878*.
- [38] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," 2020, *arXiv:2003.01690*.
- [39] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, "Square attack: A query-efficient black-box adversarial attack via random search," 2019, *arXiv:1912.00049*.
- [40] F. Croce and M. Hein. (2020). *Minimally Distorted Adversarial Examples With a Fast Adaptive Boundary Attack*. [Online]. Available: <https://openreview.net/forum?id=HJlxgBtWH>
- [41] F. Tramèr, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," 2020, *arXiv:2002.08347*.

- [42] T. Pang, X. Yang, Y. Dong, H. Su, and J. Zhu, "Bag of tricks for adversarial training," in *Proc. Int. Conf. Learn. Represent.*, 2021. [Online]. Available: <https://openreview.net/forum?id=Xb8xvrtB8Ce>
- [43] A. Shafahi, A. Ghiasi, F. Huang, and T. Goldstein, "Label smoothing and logit squeezing: A replacement for adversarial training?" 2019, *arXiv:1910.11585*.
- [44] C. Fu, H. Chen, N. Ruan, and W. Jia, "Label smoothing and adversarial robustness," 2020, *arXiv:2009.08233*.
- [45] H. Lee, H. Bae, and S. Yoon, "Gradient masking of label smoothing in adversarial robustness," *IEEE Access*, vol. 9, pp. 6453–6464, 2021.
- [46] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.
- [47] H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, and A. Madry, "Do adversarially robust ImageNet models transfer better?" 2020, *arXiv:2007.08489*.
- [48] T. Pang. (2019). *Mixup-Inference*. [Online]. Available: <https://github.com/P2333/Mixup-Inference>
- [49] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [50] J. Byun, H. Go, and C. Kim, "On the effectiveness of small input noise for defending against query-based black-box attacks," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 3051–3060.
- [51] Z. Qin, Y. Fan, H. Zha, and B. Wu, "Random noise defense against query-based black-box attacks," in *Proc. Adv. Neural Inf. Process. Syst.*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds. 2021. [Online]. Available: <https://openreview.net/forum?id=ZPSD4xZc6j8>
- [52] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.
- [53] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.



Muzammal Naseer received the Ph.D. degree from The Australian National University (ANU), Canberra, ACT, Australia, in 2022.

He served as a Researcher with Data61, CSIRO, Sydney, NSW, Australia, and the Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates, from 2018 to 2020. He is currently a Post-Doctoral Researcher with the Mohamed Bin Zayed University of Artificial Intelligence. He received the Competitive Postgraduate Scholarship. He has authored or coauthored at well-recognized machine learning and computer vision venues, including Neural Information Processing Systems (NeurIPS), International Conference on Learning Representations (ICLR), International Conference on Computer Vision (ICCV), and Computer Vision and Pattern Recognition (CVPR), with two oral and two spotlight presentations.

Dr. Naseer received the Gold Medal for outstanding performance in the B.Sc. degree. He received the Student Travel Award from NeurIPS in 2019.



Salman Khan (Senior Member, IEEE) received the Ph.D. degree from The University of Western Australia, Perth, WA, Australia, in 2016.

From 2016 to 2018, he was a Research Scientist with Data61, CSIRO, Sydney, NSW, Australia. He was a Senior Scientist with the Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates, from 2018 to 2020. He has been an Adjunct Lecturer with The Australian National University, Canberra, ACT, Australia, since 2016, and an Associate Professor with the Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, since 2020. His research interests include computer vision and machine learning.

Dr. Khan has served as a Program Committee Member for several premier conferences, including Computer Vision and Pattern Recognition (CVPR), International Conference on Computer Vision (ICCV), International Conference on Intelligent Robots and Systems (IROS), International Conference on Robotics and Automation (ICRA), International Conference on Learning Representations (ICLA), Neural Information Processing Systems (NeurIPS), and European Conference on Computer Vision (ECCV). His Ph.D. thesis received an honorable mention on the Dean's List Award. In 2019, he was a recipient of the Outstanding Reviewer Award at CVPR and the Best Paper Award at International Conference on Pattern Recognition Applications and Methods (ICPRAM) 2020.



Fatih Porikli received the Ph.D. degree from New York University, New York, NY, USA, in 2002.

He was a Distinguished Research Scientist with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. He was a full-tenured Professor with the Research School of Engineering, The Australian National University, Canberra, ACT, Australia, and a Chief Scientist with the Global Media Technologies Laboratory, Huawei, Santa Clara, CA, USA. He is currently the Global Lead of Perception with Qualcomm, San Diego, CA, USA. He has authored over 300 publications, co-edited two books, and invented 66 patents. His research interests include computer vision, pattern recognition, manifold learning, image enhancement, robust and sparse optimization, and online learning with commercial applications in video surveillance, car navigation, robotics, satellite, and medical systems.

Dr. Porikli was a recipient of the Research and Development 100 Scientist of the Year Award in 2006. He received five best paper awards at premier IEEE conferences and five other professional prizes. He is serving as an associate editor for several journals for the past 12 years. He has also served on the organizing committees of several flagship conferences, including International Conference on Computer Vision (ICCV), European Conference on Computer Vision (ECCV), and Computer Vision and Pattern Recognition (CVPR).



Fahad Shahbaz Khan received the M.Sc. degree in intelligent systems design from the Chalmers University of Technology, Gothenburg, Sweden, in 2008, and the Ph.D. degree in computer vision from the Autonomous University of Barcelona, Barcelona, Spain, in 2011.

He is currently a Faculty Member with the Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates, and Linköping University, Linköping, Sweden. He has authored or coauthored over 100 conference papers, journal articles, and book contributions in these areas. His research interests include a wide range of topics within computer vision and machine learning, such as object recognition, object detection, action recognition, and visual tracking.

Dr. Khan serves as a Regular Program Committee Member for leading computer vision conferences, such as Computer Vision and Pattern Recognition (CVPR), International Conference on Computer Vision (ICCV), and European Conference on Computer Vision (ECCV). He has achieved top ranks on various international challenges (Visual Object Tracking VOT: 1st 2014 and 2018, 2nd 2015, 1st 2016; VOT-TIR: 1st 2015 and 2016; OpenCV Tracking: 1st 2015; and 1st PASCAL VOC 2010). He received the Best Paper Award in the computer vision track at the IEEE International Conference on Pattern Recognition (ICPR) 2016. He has served as a Guest Editor for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. He is an Associate Editor of the *Image and Vision Computing Journal*.